[0049] It has been found that the rebalancing of a tree data structure can be advantageously deferred. More particularly, the method described with reference to FIG. 1 in which operations are divided into the update phase of step 130 and the commit phase of step 140 finds particular use in such deferred rebalancing.

[0050] With reference to FIG. 7, a tree data structure generally designated 700 is shown including a balanced portion having three levels generally designated 750, 760, and 770. Tree data structure 700 further includes an unbalanced portion having three levels generally designated 780, 785, and 790. In one aspect of the invention, the rebalancing of tree data structure 700 is advantageously deferred until the unbalanced portion of tree data structure 700 reaches a threshold number of levels.

[0051] It has been found that such deferred rebalancing minimally affects the average search of tree data structure 700. For example the maximum number of steps in a binary search of a binary tree data structure having n nodes is $\log_2 n$. Thus for a tree data structure 700 having 64K nodes, the maximum number of steps is 16. It has been found that bounding the length of the unbalanced portion of tree data structure 700 to two times the maximum number of steps in the binary search does not adversely affect the performance of the system in terms of the average

number of steps taken to find a particular node of tree data structure 700.

[0052] FIG. 8 illustrates a process, according to an embodiment of the invention, by which the unbalanced portion of tree data structure 700 is allowed to grow unbalanced until a leaf 730 is inserted at the threshold level of tree data structure 700. After this insertion occurs, the entire tree data structure 700 is rebalanced. By delaying the rebalancing, and allowing unbalanced sub-trees of length greater than one, the number of times the tree data structure 700 is rebalanced is reduced. This reduction saves the computational effort required to perform tree data structure rebalancing while still limiting the total depth of the tree data structure 700.

[0053] As shown in FIG. 8, operations are performed in a step 810 on tree data structure 700 (FIG. 9). These operations include node insertion, deletion, reading, and the like. Each time an operation is performed in step 810, computer code 220(FIG.2) determines in a step 820 if the operation was a node insertion. If the operation was not a node insertion then computer code 220 returns to step 810 to perform another operation. If the operation was a node insertion, then computer code, 220 determines, in a step 830, if the element node was inserted at the threshold level. If the element node was not inserted at the threshold level then computer code 220 returns to step 810.

If the insertion occurred at the threshold level then tree data structure 700 (FIG. 9) is rebalanced in a step 840. After rebalancing step 840, computer code 220 returns to step 810 and performs further operations on the tree data structure 700.

[0054] In an alternative embodiment, the threshold level compared with the insertion level in step 830 is calculated by adding a constant, such as eight, to the depth of the tree data structure 700 determined after rebalancing step 840. This approach allows tree data structure 700 to grow a constant number levels deeper than the minimum number of levels required to hold the elements nodes within tree data structure 700.

[0055] After the rebalancing, tree data structure 700 may be organized as tree data structure 900 shown in FIG. 9. In a preferred embodiment, the rebalancing includes use of the two phase process discussed with respect to FIGs. 1-6, that is, the division of the rebalancing operation into tasks executed in the update phase of step 130 (FIG. 1) and the commit phase of step 140. In this manner the rebalancing operation can be executed concurrently with other operations executing on the tree data structure 700.

[0056] Several embodiments are specifically illustrated and/or described herein. However, it will be appreciated that modifications and variations are covered by the above teachings